

AD-A093 263

VIRGINIA POLYTECHNIC INST AND STATE UNIV WASHINGTON --ETC F/6 9/2
TAPE DUMP AND RESTORE.(U)

AUG 80 R J ORGASS

AFOSR-79-0021

UNCLASSIFIED

VPI/SU-TM-80-5

AFOSR-TR-80-1291

NL

| n |
AD-
ACQUISIT



END
DATE
FILMED
2 81
DTIC



AFOSR-TR- 80 - 1291

SR (D)

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE
GRADUATE PROGRAM IN NORTHERN VIRGINIA

P. O. Box 17186
Washington, D. C. 20041
(703) 471-4600

AD A093263

LEVEL II

(Partial Draft)

TAPE DUMP AND RESTORE*

Richard J. Orgass

Technical Memorandum No. 80-5

August 28, 1980

DTIC
ELECT
S DEC 23 1980
F

ABSTRACT

A set of three programs to write CMS disk files on magnetic tape in a variety of formats suitable for reading on a variety of computing systems and for restoring these tape files to disk within the restrictions imposed by the University Computing Center is described. The documentation is partial because the author was unable to complete the document before leaving Virginia Tech.

DDC FILE COPY

- * Research sponsored by the Air Force Office of Scientific Research, Air Force Systems Command, under Grant No. AFOSR-79-0021. The United States Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon.

Approved for public release;
distribution unlimited.

Copyright, 1980

by

Richard J. Orgass

General permission to republish, but not for profit, all or part of this report is granted, provided that the copyright notice is given and that reference is made to the publication (Technical Memorandum No. 80-5, Department of Computer Science, Graduate Program in Northern Virginia, Virginia Polytechnic Institute and State University), to its date of issue and to the fact that reprinting privileges were granted by the author.

| 1. REPORT DOCUMENTATION PAGE | | READ INSTRUCTIONS BEFORE COMPLETING FORM | | | | | | | | | | | | | | | | | | | | | |
|--|--|--|--|---------------|--|------------|-------------------------------------|----------|--------------------------|-------------|--------------------------|---------------|--|----|--|---------------|--|--------------------|--|------|-------------------------|----------|--|
| 2. REPORT NUMBER AFOSR-TR-80-1291 | 3. GOVT ACCESSION NO. AD-HC93263 | 4. RECIPIENT'S CATALOG NUMBER | | | | | | | | | | | | | | | | | | | | | |
| 5. TITLE (and Subtitle) TAPE DUMP AND RESTORE | | 6. TYPE OF REPORT & PERIOD COVERED INTERIM | | | | | | | | | | | | | | | | | | | | | |
| 7. AUTHOR(s) Richard J. Orgass | | 8. CONTRACT OR GRANT NUMBER(s) AFOSR-79-0021 | | | | | | | | | | | | | | | | | | | | | |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS Dept. of Computer Science Virginia Polytech. Inst. and State University | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61102F 16 2304 A5 | | | | | | | | | | | | | | | | | | | | | |
| 11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Office of Scientific Research/ NM Bolling AFB, Washington, D.C. 20332 | | 12. REPORT DATE 28 August 1980 | | | | | | | | | | | | | | | | | | | | | |
| 13. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) RTI | | 14. NUMBER OF PAGES 18 | | | | | | | | | | | | | | | | | | | | | |
| 15. SECURITY CLASS. (of this report) UNCLASSIFIED | | 16. DECLASSIFICATION/DOWNGRADING SCHEDULE | | | | | | | | | | | | | | | | | | | | | |
| 17. DISTRIBUTION STATEMENT (of this Report) Approved for public release, distribution unlimited. | | | | | | | | | | | | | | | | | | | | | | | |
| 18. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) <table border="1"><tr><td colspan="2">Accession For</td></tr><tr><td>NTIS GRA&I</td><td><input checked="" type="checkbox"/></td></tr><tr><td>DTIC TAB</td><td><input type="checkbox"/></td></tr><tr><td>Unannounced</td><td><input type="checkbox"/></td></tr><tr><td colspan="2">Justification</td></tr><tr><td colspan="2">By</td></tr><tr><td colspan="2">Distribution/</td></tr><tr><td colspan="2">Availability Codes</td></tr><tr><td>Dist</td><td>Avail and/or Special</td></tr><tr><td>A</td><td></td></tr></table> | | | | Accession For | | NTIS GRA&I | <input checked="" type="checkbox"/> | DTIC TAB | <input type="checkbox"/> | Unannounced | <input type="checkbox"/> | Justification | | By | | Distribution/ | | Availability Codes | | Dist | Avail and/or Special | A | |
| Accession For | | | | | | | | | | | | | | | | | | | | | | | |
| NTIS GRA&I | <input checked="" type="checkbox"/> | | | | | | | | | | | | | | | | | | | | | | |
| DTIC TAB | <input type="checkbox"/> | | | | | | | | | | | | | | | | | | | | | | |
| Unannounced | <input type="checkbox"/> | | | | | | | | | | | | | | | | | | | | | | |
| Justification | | | | | | | | | | | | | | | | | | | | | | | |
| By | | | | | | | | | | | | | | | | | | | | | | | |
| Distribution/ | | | | | | | | | | | | | | | | | | | | | | | |
| Availability Codes | | | | | | | | | | | | | | | | | | | | | | | |
| Dist | Avail and/or Special | | | | | | | | | | | | | | | | | | | | | | |
| A | | | | | | | | | | | | | | | | | | | | | | | |
| 19. SUPPLEMENTARY NOTES | | | | | | | | | | | | | | | | | | | | | | | |
| 20. KEY WORDS (Continue on reverse side if necessary and identify by block number) | | | | | | | | | | | | | | | | | | | | | | | |
| 21. ABSTRACT (Continue on reverse side if necessary and identify by block number) <p>A set of three programs to write CMS disk files on magnetic tape in a variety of formats suitable for reading on a variety of computing systems and for restoring these tape files to disk within the restrictions imposed by the University Computing Center is described. The documentation is partial because the author was unable to complete the document before leaving Virginia Tech.</p> <p>4/1/73</p> | | | | | | | | | | | | | | | | | | | | | | | |

NOTE:

This is a partial DRAFT of a technical memorandum describing a program to dump files to tape and to restore files from tape. The entire report could not be completed before the author (RJ Orgass) left Virginia Tech.

Note particularly the comments at the end of the document.

TAPE DUMP AND RESTORE

1. Problem Statement

The set of programs described in this report were designed to facilitate the use of backup tapes with VM/CMS as provided at Virginia Tech and to make it possible to transfer files from this system to other computer systems, possibly machines using the ASCII character set.

An interactive user will frequently accumulate a large number of files on the system disks. A fair number of these files are not used actively and a user would be quite happy to store them on tape if it were easy to retrieve this files from tape. The programs described here provide a reasonably convenient mechanism for dumping files to tape and then restoring them to disk, possibly with a different fileid. The convenience of tape restoration is, however, limited by the Computing Center's restriction on the use of tapes in VM -- it is necessary to run a batch job to retrieve files.

The need to maintain records of the contents of a tape will often discourage users from taking advantage of a backup facility. It is simply too easy to lose the information. The programs described here rely on a directory that is written on the tape. A user is expected to remember only three things: (1) The number of distinct files written on the tape. (2) The blocking factor of the tape. (3) The recording density of the tape. If this information should be lost, it can be retrieved from the tape itself by the computing center.

Magnetic tape is a reasonably reliable storage medium but it is possible to lose data because of dirt on the tape or other problems which make it impossible to read a fragment of the reel.

When a group of files is written onto a tape they are repeated until the end of the reel so that damage to one copy of a file does not cause a loss of data. Almost the whole tape must be destroyed if it is to become impossible to retrieve a file.

If one wishes to transfer files to another VM installation, particularly one which does not have an MVS facility, a simple tape format that makes it possible to easily reconstruct the original files is needed if the transfer is to proceed successfully. The tape dump program provides such a format.

If one wishes to transfer files to a different computing system which uses the ASCII character set it is important to write ASCII tapes and to provide a simple format that can be read on other machines. The tape dump program is capable of writing 8-bit ASCII tapes that can be read on other computers.

The University Computing Center does not permit VM/CMS users to mount tapes on the VM machine and, therefore, tape writing and reading must be done using MVS. The tape dump program described here actually spools one or more MVS jobs to the virtual punch and these jobs actually write the tape. Similarly, reading files from a dump tape is accomplished in two steps. First an MVS job to read the files from the tape is created. When this job delivers the output to the virtual reader, a program to restore the files to disk is executed. The need to use MVS is almost transparent to the user; the interface appears mainly by requiring information such as job name, account number and longkey. In addition, it is necessary to examine MVS output to determine that a tape has been written successfully. One would wish for a simpler procedure using the tape capabilities of VM/CMS.

The next two sections provide directions for writing dump tapes and for retrieving files from such tapes. The next section provides a reasonably detailed description of the tape format and the last section provides information for those who wish to modify or maintain these programs.

2. Dumping Files to Tape

This section contains a set of instructions for writing files to tape using the dump program.

The first step is to secure a magnetic tape and check it into the Computing Center Tape Library. When you check in the tape, ask the tape librarian to clean and certify the tape and then to initialize it as a NON LABELED tape. This cleaning and certification is a good way to prevent a loss of data because of a defective tape. Many of us often think that a tape is always perfectly usable -- it's simply not true.

When you check in your tape, the librarian will provide you with a volume id, the name by which the tape is known in the computing center. You will need this volume id when you wish to read or write the tape. When the tape is available in the tape library, you are ready to write files onto the tape.

The tape dump program will ask a number of questions while it is in execution. If a question is followed by some text enclosed in slashes (/), this text will be your response if you simply respond by hitting the return key on your terminal.

Now that you are ready to write the tape, log in to CMS and then link to the Computer Science Library by typing the following commands:

```
cp link csdulses 191 250 rr all
access 250 g/a
```

This will make the dump and restore programs as well as other utilities available to you.

Before actually dumping files to tape, it is necessary to prepare a list of the fileids and other information about the files. The CMS command LF provides an easy way to do this. Simply enter the command

```
lf * * a (disk
```

When this command has been executed a file, CMS EXEC, will be on your disk. This file contains one line for each file in your directory. If you don't want to dump all of your files to tape, use your favorite editor to delete the lines in this file that refer to files that you do not wish to dump to tape. Once this editing is completed, it's a good idea to rename the file to some other name. This way, if you should use the LF command again the list of files for dumping won't be destroyed by accident. The command

```
rename cms exec a tpdmp dir a
```

will do this job just fine and your list of files will be in a file named TPDMP DIR. Here is an example of such a file:

| | | | | | | | | | |
|--------|---------|----|---|-----|----|---|---------|----------|--------|
| FIG353 | BOOK | A1 | F | 80 | 21 | 3 | 6/25/80 | 13:57:00 | ORGASS |
| HEAD | SCRIPT | A1 | V | 48 | 12 | 1 | 2/11/80 | 13:25:00 | ORGASS |
| FIG354 | BOOK | A1 | F | 80 | 13 | 2 | 6/26/80 | 17:09:00 | ORGASS |
| TSDAT | LISTING | A1 | F | 133 | 51 | 9 | 7/10/80 | 7:33:00 | ORGASS |

You need only concern yourself with the file name, <fn>, the file type, <ft>, and the file mode <fm> information. The dump program does, however, need the other information.

Now that you have prepared a list of files to be dumped to tape you are ready to begin the process of creating an MVS job to write your files onto a tape. If you enter the CMS command

tpdmp

this will initiate a dialog with the program in which you describe the job that you want done. It is necessary to provide some information to the program so that the files you wanted to write can be written on your tape.

The first prompt is:

Job Name:

your answer should be the name by which your MVS job is to be known. It's a good idea to use the letter b followed by your bin number and followed by three letters. For example, I usually provide

bl007org

as my job name. After your response, hit the return key. The line might look like this just as you hit return.

Job Name: bl007org

The next prompt from the program asks for your computing center account number. This number is printed in the accounting information when you log off and can also be obtained from your account supervisor. The prompt is:

Account number:

For this discussion, let's use 12345 as the account number. After your response, the line will look like this:

Account number: 12345

The next prompt asks for the volume id of your tape. Suppose that when you checked in your tape at the computing center the volume id ABC was assigned to the tape. The prompt with the appropriate response would be:

Volume id: abc

The next prompt asks for the tape density. If your tape is certified for 1600 bpi it is best to write at this density because less tape is required to contain your files. The prompt is:

Tape density: /1600/:

If you are going to write your tape at 1600 bpi, you can select the default answer by simply hitting the return key. However, if your tape was only certified for 800 bpi, you would respond 800 and the line would look like this just as you are ready to hit return:

Tape density: /1600/: 800

If one were to write 80 column records on a 1600 bpi tape one would have the following rather disturbing state of affairs. Each record of data would require 1/20 of an inch of tape and there would be a one inch record gap between tape records. This would mean that only 1/20th of the tape would contain data. Therefore, it is desirable to block logical records into longer tape records. A generally useful blocking factor is 20 logical records per tape record. This means that half of the tape is filled with data and it is still fairly easy to retrieve files from the tape. You may find that there is an advantage to using a larger blocking factor as this may save tape. Let's suppose that the tape is to be written with a blocking factor of 20. When TPDMP prompts:

Blocking factor: /20/:

simply hit return. If you want to use a different blocking factor just enter the value you wish to use.

TPDMP can be used to write tapes in either the EBCDIC or ASCII character set and tapes can be restored in either character set. If you are using TPDMP to back up files that you are planning to read back into a VM or MVS system, it is best to write your tape in the EBCDIC character set. On the other hand, if you are writing a tape to transfer files to a computing system that uses the ASCII character set, then you should write the tape in ASCII. Examples of machines that use ASCII are: DECSYSTEM-10, DEC VAX, HP 2000, HP 3000. When TPDMP prompts

Tape character set: /ASCII/:

simply hit return if you are writing a tape to be sent to another machine or respond ebcdic if you are writing a tape for use with an IBM system or a Xerox Sigma system. In this case, the line would look like this just before you hit return:

Tape character set: /ASCII/: ebcdic

At this point you have completely described the tape that is to be written. The next step is to specify the files that are to be written on the tape. Before running TPDMP, you created a directory file. In the above example, this file was called TPDMP DIR. The next prompt from TPDMP asks for the name of this file. The prompt is:

Directory file: /CMS EXEC/:

In the example we are following, the line would look like this just before you hit return:

Directory file: /CMS EXEC/: tpdmp dir

TPDMP writes a report file while your files are being spooled to tape. The next prompt requests the name of this file:

Report output file: /tpdmp report/:

If the report is to be written to file TPDMP REPORT, simply hit return. If the report is to be written to some other file, simply enter the file name and file type of the report file and then hit return. It's a good idea to use the same file name for the directory and report file. In this example, the directory is in file TPDMP DIR and the report is to be written to file TPDMP REPORT.

At this point, TPDMP will start processing the input directory in preparation for writing your files to tape. TPDMP cannot write MODULE files to tape because of a SIMULA language restriction: The record length of a MODULE file is greater than the longest allowable text object in the IBM SIMULA implementation. If your directory contains an entry for a MODULE file, this file will be rejected and a message to this effect will be printed on the terminal at this point. The same message is written to the report file so you need not save the terminal transcript.

When you are writing an ASCII tape, it simply doesn't make sense to write files other than character files on the tape. Therefore, when writing an ASCII tape, TPDMP will reject files whose file type is one of the following: VSAPLWS, MODULE, TEXT LIB. A message indicating that the file has been rejected is written to the terminal and to the report file.

After these messages (if any), TPDMP will print a summary of your request. For the directory file shown above, this message will be:

A directory file followed by 4 data files
will be written on tape ABC in the EBCDIC character set.
with a blocking factor of 20.

After this summary of your request, TPDMP will ask you if you want to display the directory of the tape you are going to write:

Display the tape director /y/:

If you simply hit return, the following would be printed on your terminal:

EBCDIC
ABC
ORGASS

1 AUG 1980
 2:18:36.00
 5 files.
 20 eighty column logical records per tape record.

| | | | | | | | |
|---|----------------------|---------|----|----|-----|---|-----|
| 1 | ** tape directory ** | | 13 | 80 | F | 1 | 13 |
| 2 | FIG353 | BOOK | A1 | 21 | 80 | F | 2 |
| 3 | HEAD | SCRIPT | A1 | 12 | 48 | V | 1 |
| 4 | FIG354 | BOOK | A1 | 13 | 80 | F | 1 |
| 5 | TSDAT | LISTING | A1 | 51 | 133 | F | 6 |
| | | | | | | | 102 |

This text is exactly the same as the contents of the first file on your tape when it is written. The entries in the directory can be described as follows.

The first line indicates the character set in which the tape is written. This line is used by TPRES to correctly restore your files. The second line is the volume id of the tape and the third line is the name of the user who wrote the tape. The fourth and fifth lines give the date and time of tape writing. The sixth line gives the number of files on the tape and the seventh line gives the blocking factor of the tape. The line of dashes serves to separate this preliminary information from the directory entries.

Each subsequent line describes a file on the tape. The first number is the file number on the tape of the file whose fileid follows this number. The next number is the number of records in the file and the number just before the letter is the record length of the file. The letter F indicates a fixed length file and the letter V indicates a variable length file. The next number is the number of tape blocks to be written for the file and the last number indicates the number of records that are needed to write the file on tape.

If a file has a record length greater than 80, each line of the file must be divided into a sequence of 80 column records so that the data can be passed to MVS which accepts only 80 column records. This decomposition is performed by TPDMP and the files are correctly reassembled by TPRES.

For example, in the above directory the third file on the tape has fileid HEAD SCRIPT A1 and consists of 12 variable length records and the longest record in the file contains 48 characters. With a blocking factor of 20, 1 tape block is used to contain the file and 13 80 column records contain data from the file.

In contrast, the fifth file on the tape has a record length of 133 characters. In this case, each of the 51 records in the file occupy two 80 column records and 102 records in six blocks are needed to contain the file.

You need not concern yourself with the details of the directory entries, this is all taken care of by TPDMP and TPRES but it might be of interest.

After the directory has been displayed, you have an opportunity to decide to terminate the tape dump. For example, you might have discovered that some files are missing from the directory or that there are files in the directory that you do not wish to write to tape. After displaying the directory, the prompt

Do you want to write the tap /y/:

is printed. If you want to continue with the tape dump, just hit return; otherwise enter n and then hit return.

After this, TPDMP will compute the approximate length of tape required to write your files. The message text for the above example is:

Approximately 2 feet of tape are needed to write these files.

At this point, you should check that the tape required is less than the length of your tape. In this example, there is, of course, enough room on any tape. However, if you are writing a large number of files, you might find that 2700 feet of tape are required to write the files and that only 2400 feet of tape are available. If this happens, you should terminate the tape dump and divide your files into two groups so that they fit on two reels of tape. To permit a termination of tape dumping at this point, TPDMP again asks if you want to write the tape. To continue, simply hit return.

The next decision requires a bit of explanation. Magnetic tape is a reasonably reliable storage medium but it is possible to encounter bad spots on the tape for a variety of reasons. Yes, this can happen even if you have the tape cleaned and certified just before writing the tape. Therefore, it is a good idea to write multiple copies of files on a tape. If you do this, then a defective spot on the tape only destroys one copy of your file not the only existing copy.

When writing a set of files on a tape, I like to write at least three copies to avoid the possibility of lost data. I call each copy of the set of files a save set. The next prompt from TPDMP asks for the number of repetitions of your files to be written. Before answering, a small computation is needed. You should divide the length of your tape by the length of tape needed to write your files. Take the larger of smaller of 3 and the result of this computation. This will be your response to the next prompt:

Number of repetitions to be written: /3/:

If you are particularly concerned about preserving particularly important data, you might want to write more than three save sets on the tape as a precaution. Do remember: It is a good idea to write at least 3 copies of your files.

Finally, all of the specifications for the tape dump have been completed and TPDMP is ready to write your files to MVS to cause a tape to be written. There is, however, one remaining prompt:

Output device: /PUN:/:

If you select the default option by hitting return, the MVS job to write your tape will be sent directly to MVS without further intervention. However, if you should want to look at the job file before sending it to MVS, you can enter a <fn> <ft> at this point. However, be careful -- the output files are very large and your disk might well overflow. Also, there is no error checking on this response and if you give an incorrect <fn> <ft> you will get a lot of strange output.

While writing your files to MVS for transfer to tape, TPDMP will advise you of its progress. Here is a sample of the output for the above example:

The files dumped for save set 1 are:

| | |
|----------------------|----------------------|
| ** tape directory ** | 20 records written. |
| FIG353 BOOK A1 | 40 records written. |
| HEAD SCRIPT A1 | 20 records written. |
| FIG354 BOOK A1 | 20 records written. |
| TSDAT LISTING A1 | 120 records written. |

This kind of output, which also appears in the report file will be printed for each save set written on the tape.

While your files are being written to tape, you may get some other messages. These messages are explained in Appendix A.

After all of your files have been spooled, TPDMP will print a summary on the terminal and in the report file. The summary looks something like this:

A job to write 1 copies of 5 files (including directories) has been spooled to disk.

Each copy of this set of files is called a SAVE SET.

Approximately 17600 bytes of data will be written on the tape as 220 80 column records.

Each save set consists of 17600 bytes or 220 80 column records.

Check the MVS output listing BEFORE deleting the files.

[TPDMP: End of processing.]

At this point, you might be inclined to assume that everything is done and that your files are safely written on tape. This is CERTAINLY NOT THE CASE. It is important to check to see that your files have been correctly written before deleting them from the disk.

After TPDMP completes processing, one or more jobs will have been submitted to MVS. After these jobs finish processing, you will find a large number of files in your virtual reader. These files are the output from MVS and it is, unfortunately, necessary to examine these files to make sure that your job ran correctly. A good way to do this is to read these files onto your disk as a single file and then print the file on a line printer. This can be done as follows:

```
cp spool rdr cl a cont
read mvs file
print mvs file
```

After your listing is on the printer, you can delete the file from your directory. You should examine this listing to confirm that your tape was written successfully. If you don't know how to interpret the vast volume of output from MVS, ask user services to help you. WHATEVER YOU DO, DON'T DELETE YOUR DISK FILES UNTIL YOU HAVE VERIFIED THE OUTPUT FROM MVS. YOU COULD LOSE YOUR FILES.

There is one other task that you might want to do to provide yourself with a permanent written record of the contents of your tape: print the report file. This report file is written to be printed on terminals or line printers using the program SPL. To begin printing your report file, enter the CMS command:

```
spl
```

The output from this program will provide an explanation of the input that is expected. If you don't know the answer to a prompt, simply enter a question mark (?), an explanation will be printed.

This report may be useful when maintaining your library of tapes but it is not needed to restore your files from tape: all of the information (except the number of files, blocking factor and tape density) are recorded on the tape itself.

3. Restoring Files from Tape

Since ordinary users are not permitted to use the magnetic tape capabilities of VM/CMS, restoring files from tape is a two step process. The first step is to create an MVS job to read the

files from the tape and the second step is to read the files from the virtual reader and restore them to disk. Both of these tasks are accomplished by running programs.

The first step in restoring files from tape is to check the tape into the computing center tape library. When the tape is checked in, a volume id will be assigned to the tape. If possible, this should be the same volume id that was used when the tape was written but this is not necessary -- any volume id is acceptable.

After your tape is available in the computing center tape library, log into CMS and link to the Computer Science library by typing the commands:

```
cp link csdulses 191 250 rr all
access 250 g/a
```

When this is done, the tape restoration programs and other utilities are available for use.

The first step is to initiate an MVS job to read your files from your tape. This is done by entering the CMS command

```
getfiles
```

This will initiate a dialog with a program to specify the work that is to be done.

The first prompt from the program requests a job name. My usual job name is bl007org and the terminal line looks like this just before I hit return:

```
Job Name: bl007org
```

The second prompt asks for your computing center account number. If your account number is 12345, the terminal line looks like this just before hitting return:

```
Account number: 12345
```

The third prompt asks for the longkey associated with this account. If the long key is xx, then the next line of the terminal dialog looks like this just before hitting return:

```
Long key: xx
```

The next prompts ask for information about the tape that was written by TPDMP. The first prompt asks for the volume id. Following the example in Section 2, the volume id is ABC and the terminal line looks like this just before hitting return:

```
Volume id: abc
```

The next prompt asks for the tape density. If you selected the default answer when writing the tape, you should select the default answer here too; otherwise give the same answer you gave when writing the tape. To select the default answer, simply hit return when the prompt

Tape density: /1600/:

is written on your terminal. Otherwise, give the same answer as you gave when writing the tape and then hit return.

The next prompt asks for the blocking factor. Again, if you selected the default blocking factor when writing the tape, select it here too. If you gave a different answer, give the answer you gave when writing the tape. If you select the default, the terminal line just before you hit return looks like this:

Blocking factor: /20/:

The next prompt asks for the number of different files that were saved by TPDMP. In the example that we used for TPDMP, five files were saved, one directory file and five data files. For this files were saved, one directory file and five data files. To retrieve these files, the terminal line would look like this just before hitting return:

Number of files saved by TPDMP: /20/: 5

The next prompt asks for the save set to be read. It's more economical to use the first save set but if something goes wrong using the first save set a later save set can be used. Let's try for the first save set and the line would look like this just before hitting return:

Save set to be read: /1/:

If the restoration from this save set failed, one could try again by entering a save set number before hitting return. Note that this number must not be larger than the number of save sets written. If you give a save set number that is larger than the number of save sets written on the tape, your tape will run off the reel and you will get a rather nasty message from the operator -- quite properly, I might add.

At this point, GETFILES is ready to write the MVS job to retrieve your files. The prompt:

Output device: /PUN:/

will appear. Simply hit return. If you really want to examine the text of the job that will be submitted to MVS you can enter a <fn> <ft> here and then look at the disk file and punch it to MVS but it's really not at all interesting.

Continuing with the example from Section 2, after this the message:

An MVS job to read files 1 to 5 from tape ABC has been written.

When the resulting files appear in your virtual reader, run TPRES.

[GETFILES: End of execution.]

you have finished with this step. At this point, you might as well log off and do something useful while waiting for your MVS output. When the MVS job has been run, you will find a large supply of files in your virtual reader. When these files are collected and your MVS job has finished execution, it's time to run TPRES. The directions follow.

Link to the Computer Science library by entering the commands:

```
cp link csdulses 191 250 rr all
access 250 g/a
```

This makes the tape restore program as well as other utilities available for your use.

To initiate the tape restoration, enter the CMS command:

```
tpres
```

This will initiate a dialog concerning the restoration of your files. The first prompt is:

```
Input file: /RDR:/:
```

Select the default answer to this prompt by entering return. If you had previously spooled the punch files from your reader to disk, you could respond with the <fn> <ft> of the file containing these reader files. This is generally a bad way to proceed because you will have two copies of your files on disk at the same time and there may well not be enough space.

TPRES also writes a report file on disk so that it is not necessary to save a copy of your terminal transcript. The report really isn't needed but it's rather pleasant to feel that a record is being kept while files are restored. If you are happy with the result of the restoration, you can delete the report file but if there are questions, it's good to have the report file available. TPRES prompts for the name of the report file:

```
Report file: /tpres report/:
```

To select this report file name, simply hit return; otherwise give the <fn> <ft> of the file that is to contain the report.

After this prompt, TPRES will ask if you wish to have the tape directory displayed on your terminal. Even if you answer no, the directory will be written to the report file. The prompt is:

Display the tape director /y/:

If you simply hit return, the tape directory will be written on your terminal. If you don't want to look at the directory, enter n and then hit return.

After this, restoration of files begins. For each file, you will be asked if you wish to restore the file. Using the example directory from Section 2, the transcript might appear as follows:

Restore FIG353 BOOK A /y/:

If you just hit return, the next prompt is:

New name: /FIG353 BOOK A1/:

If you hit return, the file will be restored with the fileid that it had when the tape was written. On the other hand, if you enter a new <fn> <ft> [<fm>], the file will be restored with this new name. For example, if the line looked like this before hitting return:

New name: /FIG353 BOOK A1/: fig353 n

then the file would be restored with fileid FIG353 N A1.

On the other hand, if you answered no to the question about restoring the file, i.e.,

Restore FIG353 BOOK A /y/: n

Then the message

FIG 353 BOOK A1 skipped.

would appear on your terminal.

In either case, the next prompt would concern the following file, i.e.,

Restore HEAD SCRIPT A /y/:

and you would proceed in the same way.

This dialog continues until all of the files on the tape have been considered. After the last file has been either restored or skipped, the message

[TPRES: End of file restoration.]

is printed on the terminal.

When this step is reached, the files have been restored but there is a bit of housekeeping to be done. Your virtual reader still contains a rather large number of files from MVS that you don't want to examine at all. Get rid of these file by entering the command

```
cp purge rdr all
```

and these uninteresting files will disappear.

One remaining detail concerning the restoration procedure. At some point in the restoration, you may decide that you have restored all of your files but there are still more files to be considered. You can abort this consideration of remaining files quite easily if you are using an ASCII terminal. When one of the prompts is printed, hold down the key labeled CNTRL while you hit the C key and then hit return. This will terminate execution of TPRES. It's not a good idea to use <break> hx or <break> ht to get out of TPRES; funny things happen when you terminate in this way. The ^C sequence is much better!

NOTES TO FUTURE USERS/MAINTAINERS

Design Changes

At the time this draft was written, the author was unaware of the MVS restriction of no more than 255 steps per job.

The program TPDMP has been modified so that all jobs written by TPDMP satisfy this restriction. However, for tapes containing a save set of more than 255 files, the output from GETFILES will fail to satisfy this MVS restriction and the resulting job will not execute. A trivial modification to GETFILES to count the number of steps written and to begin another job is required.

When the documentation was written, it was assumed that the MVS output could be returned to the VM virtual reader. In the course of solving a number of problems related to this form of MVS output, TPDMP was revised to write printed output to a remote station. With the present structure of TPDMP, the MVS output could be sent to a VM virtual reader again. The program should be changed to again send this output to the virtual reader as this output saves a substantial amount of paper by making it possible to use edit to check the condition codes of each step instead of printing a listing and manually examining the condition codes. One might even be able to write an EXEC that reads the MVS output onto a tdisk and then prints a message only when a condition code other than zero is found.

The documentation implies that the MVS job(s) will start immediately. This was found to be imprudent for a rather strange reason. The current load on VM/CMS is so heavy that it is possible for MVS to complete the execution of one job before the next job has been written by VM/CMS. If this occurs, the next job will never be started because MVS will not be able to find the job when it is released for execution. Therefore, even the first job is written as a hold job. It must be released for execution by sending a message to the operator.

TPDUMP should be modified to write a small MVS job to start the first job in the sequence after the last job has been written. This will restore the automatic job start without the risk of a broken chain.

Required Files

The following files are needed to compile/load the programs described above. All of these files are on the 191 disk of userid CSDULLES.

PREFIX SIMULA
POSTFIX SIMULA
SIM EXEC
LIBSIM TXTLIB
MVSLIB TXTLIB (SIMLIB TXTLIB Y can also be used)

The SIM EXEC is a compile, load, genmod and go exec that includes a number of jcl lines that are required to compile the program.

Before executing the SIM EXEC (but after linking to the 191 of CSDULLES as a read only extension of the A disk) the command

global txtlib libsim mvslib

or

global txtlib libsim simlib

must be executed.

A draft help file for TPDMP has been created. It does not follow the conventions of the current help system. It is file TPDMP HELPCMS.

Modules for TPDMP, GETFILES and TPRES are on the 191 disk of CSDULLES.

The SIMULA source for TPDMP, GETFILES and TPRES are in file SIMUTIL GENLIB on the 192 disk of userid CSDULLES.

This document is in DOCUTIL GENLIB on the 192 disk of CSDULLES as file TM805 SCRIPT. This file imbeds file RATFOR0 SCRIPT which is part of the same genlib and is also on the 191 disk of userid SUE.

File Formats

The source files for these programs are LRECL 80, RECFM F as required by the SIMULA compiler. However, they differ from input files to compilers such as PL/I and Fortran as follows:

The files contain tab characters (which are treated as blanks by the SIMULA compiler) to provide visually longer lines. These files are in upper and lower case and the case of the text pro-

vides useful information to the reader. These files can be printed using the program SPL; the module is on the 191 disk of CSDULLES and the prompts have help associated with them; simply enter a question mark if you don't know what to answer.

The exec TEDM EXEC on the 191 disk of CSDULLES provides an entry to EDIT with the appropriate settings to edit files that contain tabs as well as other obscure details. It is recommended for editing these programs.

When using SPL to a hardcopy terminal that does not have tabs, the file TABS TEXT on the 191 disk of CSDULLES should be free-loaded before executing SPL. For further details, see STTY EXEC on the 191 disk of CSDULLES.